



Ulaz i izlaz

7.1 Standardni ulaz i izlaz

```
int getchar(void)
```

```
EOF <stdio.h>
```

```
prog <ulazdat
```

```
drugiprogram I prog
```

```
int putchar(int)
```

```
EOF
```

```
prog >izlazdat
```

```
prog I drugiprogram
```

```
#include <stdio.h>
```

```
< >
```

```
UNIX
```

```
/usr/include
```

```
#include <stdio.h>
#include <ctype.h>
main()          /* lower: convert input to lower case*/
{
    int c;

    while ((c = getchar()) != EOF)
        putchar(tolower(c));
    return 0;
}
```

7.2 Formatizovan izlaz - printf

```
int printf(char *format, arg1, arg2, ...)
```

TABELA 7.1 OSNOVNE KONVERZIJE FUNKCIJE printf

ZNAK	TIP ARGUMENTA; PRIKAZUJE SE KAO
d, i	int; dekadni broj.
o	int; neoznačeni oktalni broj (bez vodeće nule).
x, X	int; neoznačeni heksadekadni broj (bez vodećih 0x ili 0X), koristeći abcdef ili ABCDEF za 10, 11, ..., 15.
u	int; neoznačeni dekadni broj.
c	int; jedan znak.
s	char *; prikazuje znakove iz stringa do znaka '\0' ili do broja znakova datih preciznošću.
f	double; [-]m.dddddd, gde je broj cifara <i>d</i> određen preciznošću (podrazumeva se 6).
e, E	double; [-]m.dddddde±xx ili [-]m.dddddde±xx, gde je broj cifara <i>d</i> određen preciznošću (podrazumeva se 6).
g, G	double; koristi se %e ili %E ako je eksponent manji od -4 ili veći od ili jednak preciznosti; inače se koristi %f. Nule i decimalna tačka na kraju se ne prikazuju.
p	void *; pokazivač (prikazivanje zavisi od implementacije).
%	argument se ne konvertuje; prikazuje se %.

```
printf("%.s", max, s); /* za prikazivanje najviše max znakova stringa s */
```

```
:%s:      :hello, world:  
:%10s:    :hello, world:  
:%.10s:   :hello, wor:  
:%-10s:   :hello, world:  
:%.15s:   :hello, world:  
:%-15s:   :hello, world :  
:%15.10s:      : hello, wor:  
:%-15.10s:      :hello, wor :
```

```
printf(s);      /* POGREŠNO ako s sadrži % */  
printf("%.s", s); /* BEZBEDNO */
```

```
int sprintf(char *string, char *format, arg1, arg2, ...)
```

7.3 Liste argumenata promenljive dužine

```
int printf(char *fmt, ...)
```

```
void minprintf(char *fmt, ...)
```

```
#include <stdarg.h>
```

```
/* minprintf: minimal printf with variable argument list */
void minprintf(char *fmt, ...)
{
    va_list ap; /* points to each unnamed arg in turn */
    char *p, *sval;
    int ival;
    double dval;
    /* make ap point to 1st unnamed arg */
    va_start(ap, fmt);
    for (p = fmt; *p; p++) {
        if (*p != '%') {
            putchar(*p);
            continue;
        }
        switch (*++p) {
            case 'd':
                ival = va_arg(ap, int);
                printf("%d", ival);
                break;
            case 'f':
                dval = va_arg(ap, double);
                printf("%f", dval);
                break;
            case 's':
                for (sval = va_arg(ap, char *); *sval; sval++)
                    putchar(*sval);
                break;
            default:
                putchar(*p);
                break;
        }
    }
    va_end(ap); /* clean up when done */
}
```

7.4 Formatizovan ulaz - scanf

```
int scanf(char *format, ...)
```

```
int sscanf(char *string, char *format, arg1, arg2, ...)
```

TABELA 7.2 OSNOVNE KONVERZIJE FUNKCIJE scanf

ZNAK	ULAZNI PODACI; TIP ARGUMENTA
d, i	dekadni broj; int *. ceo broj; int *. Ceo broj može biti u oktalnom (vodeća 0) ili heksadekadnom zapisu (vodeći 0x ili 0X)
o	oktalni broj (sa ili bez vodeće nule); int *.
u	neoznačeni dekadni broj; unsigned int *.
x	heksadekadni ceo broj; (sa ili bez vodećih 0x ili 0X); int *.
c	znakovi; char *.
s	znakovni string (bez navodnika); char *, koji pokazuje na niz znakova dovoljno velik za smeštanje stringa i završnog znaka '\0' koji se dodaje.
e, f, g	broj sa pokretnim zarezo i opcioni znak, opciona decimalna tačka i opcioni eksponent; float *.
%	doslovni znak %; ne vrši se nikakvo dodeljivanje.

```
#include <stdio.h>
main() /* rudimentary calculator */
{
    double sum, v;
    sum = 0;
    while (scanf("%lf", &v) == 1)
        printf("\t%.2f\n", sum += v);
    return 0;
}
```

25 Dec 1988

```
int day, year;
char monthname[20];
scanf("%d %s %d", &day, monthname, &year);
```

mm/dd/yy

```
int day, month, year;
scanf("%d/%d/%d", &month, &day, &year);
```

```
while (getline(line, sizeof(line)) > 0) {
    if (sscanf(line, "%d %s %d", &day, monthname, &year) == 3)
        printf("valid: %s\n", line); /* Oblik 25 Dec 1988 */
    else if (sscanf(line, "%d/%d/%d", &month, &day, &year) == 3)
        printf("valid: %s\n", line); /* Oblik mm/dd/yy */
    else
        printf("invalid: %s\n", line); /* Pogrešan oblik */
}
```

```
scanf("%d", n); /* greška */
```

```
scanf("%d", &n);
```

7.5 Pristup datotekama

```
cat x.c y.c    /* ova komanda prikazuje sadržaj datoteka x.c i y.c na standardnom izlazu */
```

```
FILE *fp;      /* deklaracija strukture pod nazivom FILE nalazi se u <stdio.h> */  
FILE *fopen(char *name, char *mode);    /* FILE je ime tipa, a ne oznaka strukture */
```

```
fp = fopen(name, mode);
```

```
"r", "w", "a"  
"b"  
+  
NULL
```

```
int getc(FILE *fp)
```

```
int putc(int c, FILE *fp)
```

```
stdin, stdout, stderr    <stdio.h>
```

```
#define getchar()  getc(stdin)  
#define putchar(c)  putc((c), stdout)
```

```
int fscanf(FILE *fp, char *format, ...)  
int fprintf(FILE *fp, char *format, ...)
```

```
#include <stdio.h>

/* cat: concatenate files, version 1 */
main(int argc, char *argv[])
{
    FILE *fp;
    void filecopy(FILE *, FILE *)

    if (argc == 1)      /* no args; copy standard input */
        filecopy(stdin, stdout);
    else
        while(--argc > 0)
            if ((fp = fopen(*++argv, "r")) == NULL) {
                printf("cat: can't open %s\n", *argv);
                return 1;
            } else {
                filecopy(fp, stdout);
                fclose(fp);
            }
        return 0;
}

/* filecopy: copy file ifp to file ofp */
void filecopy(FILE *ifp, FILE *ofp)
{
    int c;

    while ((c = getc(ifp)) != EOF)
        putc(c, ofp);
}
```

```
int fclose(FILE *fp)
```

7.6 Rukovanje greškama - stderr i exit

```
#include <stdio.h>

/* cat: concatenate files, version 2 */
main(int argc, char *argv[])
{
    FILE *fp;
    void filecopy(FILE *, FILE *);
    char *prog = argv[0];          /* program name for errors */

    if (argc == 1)                 /* no args; copy standard input */
        filecopy(stdin, stdout);
    else
        while (--argc > 0)
            if ((fp = fopen(*++argv, "r")) == NULL) {
                fprintf(stderr, "%s: can't open %s\n", prog, *argv);
                exit(1);
            } else {
                filecopy(fp, stdout);
                fclose(fp);
            }
        if (ferror(stdout)) {
            fprintf(stderr, "%s: error writing stdout\n", prog);
            exit(2);
        }
        exit(0);
}
```

```
int ferror(FILE *fp)
```

```
int feof(FILE *fp)
```

7.7 Ulaz i izlaz po redovima

```
char *fgets(char *line, int maxline, FILE *fp)
```

```
int fputs(char *line, FILE *fp)
```

```
/* fgets: get at most n chars from iop */
char *fgets(char *s, int n, FILE *iop)
{
    register int c;
    register char *cs;

    cs = s;
    while (--n > 0 && (c = getc(iop)) != EOF)
        if ((*cs++ = c) == '\n')
            break;
    *cs = '\0';
    return (c == EOF && cs == s) ? NULL : s;
}

/* fputs: put string s on file iop */
int fputs(char *s, FILE *iop)
{
    int c;

    while (c = *s++)
        putc(c, iop);
    return ferror(iop) ? EOF : 0;
}
```

```
/* getline: read a line, return length */
int getline(char *line, int max)
{
    if (fgets(line, max, stdin) == NULL)
        return 0;
    else
        return strlen(line);
}
```

7.8 Pomoćne funkcije

7.8.1 Operacije sa stringovima

U narednoj tabeli s i t su tipa char *, a c i n su tipa int. Funkcije se nalaze u <string.h>.

strcat(s, t)	dodaje t na kraj s.
strncat(s, t, n)	dodaje n znakova t na kraj s.
strcmp(s, t)	vraća negativnu vrednost, nulu ili pozitivnu vrednost. za $s < t$, $s == t$ ili $s > t$.
strncmp(s, t, n)	isto kao strcmp, ali samo u prvih n znakova.
strcpy(s, t)	kopira t u s.
strncpy(s, t, n)	kopira najviše n znakova t u s.
strlen(s)	vraća dužinu s.
strchr(s, c)	vraća pokazivač na prvo c u s, ili NULL ako ga nema.
strrchr(s, c)	vraća pokazivač na poslednje c u s, ili NULL ako ga nema.

7.8.2 Određivanje klase znakova i konverzije

U narednoj tabeli `c` je ceo broj tipa `int` koji se može predstaviti kao `unsigned char` ili kao `EOF`. Funkcije se nalaze u `<ctype.h>`.

<code>isalpha(c)</code>	različito od 0 ako je <code>c</code> alfabetski znak, 0 ako nije
<code>isupper(c)</code>	različito od 0 ako je <code>c</code> veliko slovo, 0 ako nije.
<code>islower(c)</code>	različito od 0 ako je <code>c</code> malo slovo, 0 ako nije.
<code>isdigit(c)</code>	različito od 0 ako je <code>c</code> cifra, 0 ako nije.
<code>isalnum(c)</code>	različito od 0 ako je <code>isalpha(c)</code> ili <code>isdigit(c)</code> , 0 ako nije.
<code>isspace(c)</code>	različito od nule ako je <code>c</code> znak za razmak, tabulator, novi red, povratak na početak reda, prelazak na novu stranu, vertikalni tabulator.
<code>toupper(c)</code>	vraća <code>c</code> pretvoreno u veliko slovo.
<code>tolower(c)</code>	vraća <code>c</code> pretvoreno u malo slovo.

7.8.3 Ungetc

```
int ungetc(int c, FILE *fp)
```

dodaje znak `c` ponovo u tok `fp` i vraća bilo `c`, bilo `EOF` u slučaju greške.

7.8.4 Izvršavanje komande OS

```
system(char *); /* izvršava komandu koja se nalazi u stringu s */  
system("date"); /* u UNIX-u */
```

7.8.5 Upravljenje memorijom

```
void *malloc(size_t n)
```

```
void *calloc(size_t n, size_t size)
```

```
int *ip;  
ip = (int *) calloc(n, sizeof(int));
```

```
for (p = head; p != NULL; p = p->next) /* POGREŠNO */  
    free(p);
```

```
for (p = head; p != NULL; p = q) {  
    q = p->next;  
    free(p);  
}
```

7.8.6 Matematičke funkcije

U `<math.h>` je deklarirano više od dvadeset funkcija.

<code>sin(x)</code>	sinus od x , x u radijanima
<code>cos(x)</code>	kosinus od x , x u radijanima
<code>atan2(y, x)</code>	arkus tangens od y/x , u radijanima
<code>exp(x)</code>	eksponencijalna funkcija e^x
<code>log(x)</code>	prirodni logaritam od x ($x > 0$, osnova e)
<code>log10(x)</code>	opšti logaritam od x ($x > 0$, osnova 10)
<code>pow(x, y)</code>	x^y
<code>fabs(x)</code>	apsolutna vrednost od x

7.8.7 Generisanje slučajnih brojeva

F-ja `rand()` izračunava niz pseudoslučajnih celih brojeva
u opsegu od nule do konstante `RAND_MAX`
koja je definisana u `<stdlib.h>`

```
/* dobijanje slučajnih brojeva sa pokretnim zarezom, većih ili jednakih nuli i manjih od jedan */  
#define frand() ((double) rand() / (RAND_MAX+1.0))
```

```
srand(unsigned)
```